

EECS 5330 Image Analysis and Computer Vision

Assignment 1 “Lenna”

By Edris Amin



This project was a good refresher for my Matlab skills. It involved loading an image, gathering data on it, saturating intensity levels, adding noise, smoothing and median filtering the noisy image. The image was obtained from Dr. Salari's network folder with the following UNIX terminal command.

CP '/home/top/esalari/lenna.mat' lenna.mat

Once in the local computer the image was loaded into matlab as a grayscale image with 256 gray levels. The image was plotted on a 3 by 3 grid using the subplot() function in matlab which Joseph Pietrykowski showed me how to use.

The intensity plotting code was provided by Dr. Salari and can be found in my code under the comment "%intensity plot". This was also displayed next to the image using subplot(). The array used in the intensity plot was used to find an intensity value where 20% of the pixels were at or below that intensity. This code can be found under the comment "%lower intensity value of 20% image" the basic algorithm design was provided by Dr. Salari. After finding the lower intensity all pixels at or below that intensity value were saturated to 0 and the resulting image was output, this code can be found under the comment "%creating saturated image...". The code under the comment "%begin adding gaussian noise" contains code which adds random noise with mean intensity value 20 and variance 40 to the image. I required some assistance from Joseph Pietrykowski in writing the correct syntax for adding noise. This noisy image was referred to as the array Y and plotted using the subplot() function.

The noisy image was smoothed by setting the intensity value for each pixel to the average value of itself and its neighbors. This required some special handling of the data since not all pixels have equal number of neighbors, like the corners and edges. The code for this section can be found under the comment "%smoothing". This was handled by an if-else logic illustrated in figure 1 below. The smoothed image was referred to as the array smoothlenna.

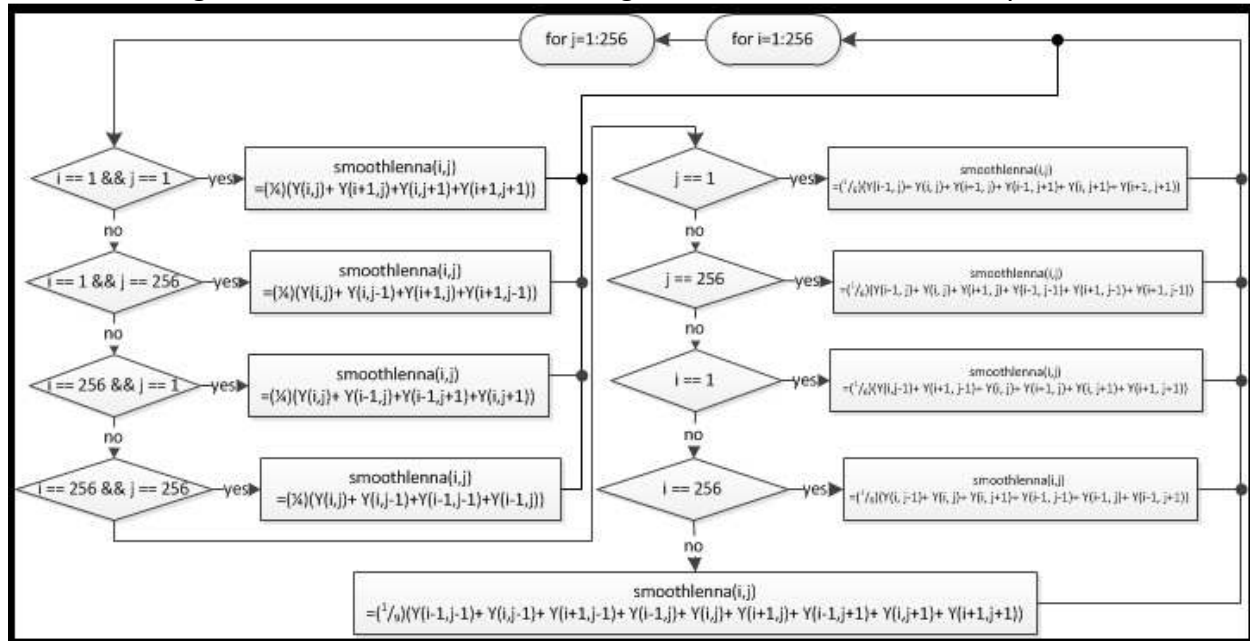


Figure 1

Producing a median filtered image also required special handling of the corner and edge pixels. This time instead of averaging the intensity the pixel was set to the median value of itself and its neighbors. This required a special sorting of intensity values. The code for this can be found under the comment “%median”. The sorting algorithm I used utilized a array of size 4 for corner pixels, size 6 for edges pixels, and size 9 for all other.

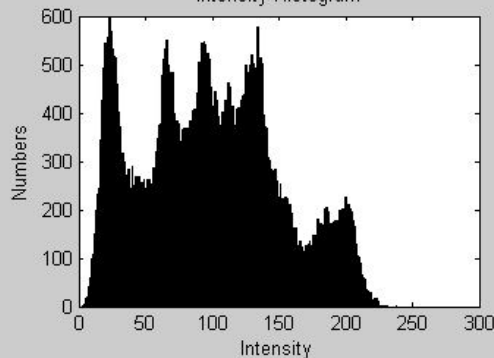
1. Set sorted variable to 0
2. Load intensity of pixels into appropriate array ([4], [6], or [9])
3. while sorted == 0
 - a. perform linear sort on intensity array
 - b. when sorted get median
 - i. for intensity array size 4 median = $(1/2) * (I_{\text{sort}(2)} + I_{\text{sort}(3)})$
 - ii. for intensity array size 6 median = $(1/2) * (I_{\text{sort}(3)} + I_{\text{sort}(4)})$
 - iii. for intensity array size 9 median = $I_{\text{sort}(5)}$
 - c. set sorted = 1
4. repeat for all pixels

The figure below shows the output from the code found on the following pages.

Original image "lenna"



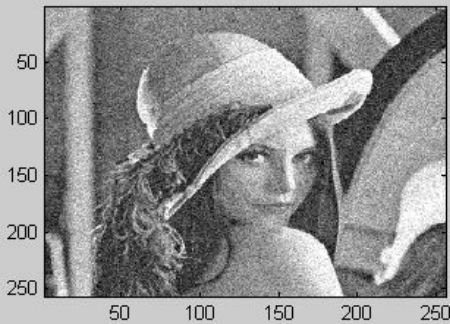
Intensity Histogram



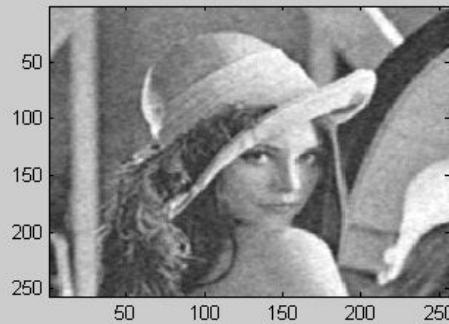
Saturated Image



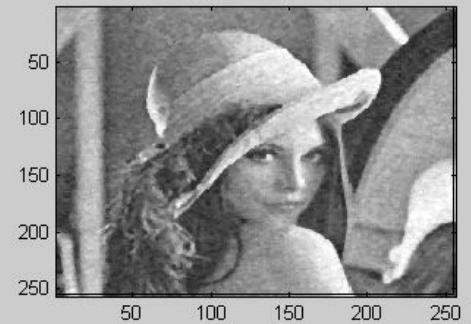
Noisy image



Smoothed Image



Median filtered Image



```

clear all

load lenna
colormap(gray(256))
subplot(3, 3, 1); image(lenna)
%The syntax for the subplot was obtained with help from Joseph Pietrykowski
title('Original image "lenna"')

%intensity plot
x=1:256;
y=zeros(1,256);
for i = 1:256;
    for j = 1:256;
        y(lenna(i,j))=y(lenna(i,j))+1;
    end
end

%subplot()
subplot(3, 3, 2); bar(x,y)%outputs figure of bar graph
%bar(x,y)%outputs figure of bar graph
title('Intensity Histogram')
xlabel('Intensity')
ylabel('Numbers')

%lower intensity value of 20% image
P = zeros(1,256);%P is the probabilities of intensities 1:256
C = 0; %sum of intensity Porb.
ival = 0; %variable for intensity value of 20%
for i=1:256;
    P(i)=100*y(i)/(256*256);
    C = P(i)+C;
    if C>=20
        ival = i;
        break
    end
end

%creating saturated image pixels where intensity <= ival will be set to 0
zero20 = zeros(256,256); %matrix for saturated image
for i= 1:256
    for j= 1:256
        if (lenna(i,j) <= ival)
            zero20(i,j) = 0;

        else
            zero20(i,j) = lenna(i,j);
        end
    end
end
subplot(3, 3, 3);
image(zero20)
title('Saturated Image');

%begin adding gaussian noise
%Joseph Pietrykowski assisted me in correcting these function to obtain
xn = randn(256);
yn = 20*xn+40;
Y=yn+lenna;
colormap(gray(256))
subplot(3, 3, 4);
image(Y)
title('Noisy image');

% %smoothing
smoothlenna = zeros(256,256); %matrix for smoothed image
for i=1:256;
    for j=1:256;
        if i == 1 && j ==1%left-top corner
            smoothlenna(i,j) = (1/4)*(Y(i,j)+ Y(i+1,j)+Y(i,j+1)+Y(i+1,j+1));
            %           1,1     2,1     1,2     2,2
        elseif i==1 && j ==256%left-bottom corner

```

```

        smoothlenna(i,j) = (1/4)*(Y(i,j)+ Y(i,j-1)+Y(i+1,j)+Y(i+1,j-1));
                                %1, 256  1, 255  2, 256  2, 255
    elseif i==256 && j==1%right-top corner
        smoothlenna(i,j) = (1/4)*(Y(i,j)+ Y(i-1,j)+Y(i-1,j+1)+Y(i,j+1));
                                %256, 1  255, 1  255, 2  256, 2
    elseif i==256 && j==256%right-bottom corner
        smoothlenna(i,j) = (1/4)*(Y(i,j)+ Y(i,j-1)+Y(i-1,j-1)+Y(i-1,j));
                                %256, 256  256, 255  255, 255  255, 256
    elseif j == 1 %top row
        smoothlenna(i,j) = (1/6)*(Y(i-1, j)+ Y(i, j)+ Y(i+1, j)+ Y(i-1, j+1)+ Y(i, j+1)+
Y(i+1, j+1));
                                %take (2,1)  1,1  2,1  3,1  1,2  2,2  3,2
    elseif j == 256 %bottom row
        smoothlenna(i,j) = (1/6)*(Y(i-1, j)+ Y(i, j)+ Y(i+1, j)+ Y(i-1, j-1)+ Y(i+1, j-1)+
Y(i+1, j-1));
                                %take (2,256)  1,256  2,256  3,256  1,255  2,255
3,255
    elseif i == 1%left row
        smoothlenna(i,j) = (1/6)*(Y(i,j-1)+ Y(i+1, j-1)+ Y(i, j)+ Y(i+1, j)+ Y(i, j+1)+
Y(i+1, j+1));
                                %take (1,2)  1,1  2,1  1,2  2,2  1,3  2,3
    elseif i == 256%right row
        smoothlenna(i,j) = (1/6)*(Y(i, j-1)+ Y(i, j)+ Y(i, j+1)+ Y(i-1, j-1)+ Y(i-1, j)+ Y(i-
1, j+1));
                                %take (256,2)  256, 1  256, 2  256,3  255,1  255, 2  255,
3
    else
        smoothlenna(i,j) = (1/9)*(Y(i-1,j-1)+ Y(i,j-1)+ Y(i+1,j-1)+ Y(i-1,j)+ Y(i,j)+
Y(i+1,j)+ Y(i-1,j+1)+ Y(i,j+1)+ Y(i+1,j+1));
    end
end
end
subplot(3, 3, 5);
image(smoothlenna)
title('Smoothed Image');

%median
medianlenna = zeros(256,256); %matrix for median image
lsort = zeros(1,9);
lsort4 = zeros(1,4);
lsort6 = zeros(1,6);
for i=1:256;
    for j=1:256;
        sorted = 0;
        if i == 1 && j ==1%left-top corner
            lsort4 = ([Y(i,j), Y(i+1,j), Y(i,j+1), Y(i+1,j+1)]);
                                % 1,1  2,1  1,2  2,2
            while(sorted == 0)
                if (lsort4(1) < lsort4(2))
                    a=lsort4(1);
                    b=lsort4(2);
                    lsort4(1) = b;
                    lsort4(2) = a;
                elseif (lsort4(2) < lsort4(3))
                    a=lsort4(2);
                    b=lsort4(3);
                    lsort4(2) = b;
                    lsort4(3) = a;
                elseif (lsort4(3) < lsort4(4))
                    a=lsort4(3);
                    b=lsort4(4);
                    lsort4(3) = b;
                    lsort4(4) = a;
                else
                    medianlenna(i,j) = (1/2)*(lsort4(2)+lsort4(3));
                    sorted = 1;
                end
            end
        elseif i==1 && j ==256%left-bottom corner
            lsort4 = ([Y(i,j), Y(i,j-1), Y(i+1,j), Y(i+1,j-1)]);

```

```

        %1, 256    1, 255    2, 256    2, 255
while(sorted == 0)
    if (lsort4(1) < lsort4(2))
        a=lsort4(1);
        b=lsort4(2);
        lsort4(1) = b;
        lsort4(2) = a;
    elseif (lsort4(2) < lsort4(3))
        a=lsort4(2);
        b=lsort4(3);
        lsort4(2) = b;
        lsort4(3) = a;
    elseif (lsort4(3) < lsort4(4))
        a=lsort4(3);
        b=lsort4(4);
        lsort4(3) = b;
        lsort4(4) = a;
    else
        medianlenna(i,j) = (1/2)*(lsort4(2)+lsort4(3));
        sorted = 1;
    end
end

elseif i==256 && j==1%right-top corner
    lsort4 = ([Y(i,j), Y(i-1,j),Y(i-1,j+1),Y(i,j+1)]);
        %256, 1    255, 1    255, 2    256, 2
while(sorted == 0)
    if (lsort4(1) < lsort4(2))
        a=lsort4(1);
        b=lsort4(2);
        lsort4(1) = b;
        lsort4(2) = a;

    elseif (lsort4(2) < lsort4(3))
        a=lsort4(2);
        b=lsort4(3);
        lsort4(2) = b;
        lsort4(3) = a;

    elseif (lsort4(3) < lsort4(4))
        a=lsort4(3);
        b=lsort4(4);
        lsort4(3) = b;
        lsort4(4) = a;

    else
        medianlenna(i,j) = (1/2)*(lsort4(2)+lsort4(3));
        sorted = 1;
    end
end

elseif i==256 && j==256%right-bottom corner
    lsort4 = ([Y(i,j), Y(i,j-1),Y(i-1,j-1),Y(i-1,j)]);
        %256,256    256,255    255,255    255,256
while(sorted == 0)
    if (lsort4(1) < lsort4(2))
        a=lsort4(1);
        b=lsort4(2);
        lsort4(1) = b;
        lsort4(2) = a;
    elseif (lsort4(2) < lsort4(3))
        a=lsort4(2);
        b=lsort4(3);
        lsort4(2) = b;
        lsort4(3) = a;
    elseif (lsort4(3) < lsort4(4))
        a=lsort4(3);
        b=lsort4(4);
        lsort4(3) = b;
        lsort4(4) = a;
    else

```

```

        medianlenna(i,j) = (1/2)*(lsort4(2)+lsort4(3));
        sorted = 1;
    end
end

elseif j == 1 %top row
    lsort6 = ([Y(i-1, j), Y(i, j), Y(i+1, j), Y(i-1, j+1), Y(i, j+1), Y(i+1, j+1)]);
    %take (2,1)          1,1          2,1          3,1          1,2          2,2
3,2
    while(sorted == 0)
        if (lsort6(1) < lsort6(2))
            a=lsort6(1);
            b=lsort6(2);
            lsort6(1) = b;
            lsort6(2) = a;
        elseif (lsort6(2) < lsort6(3))
            a=lsort6(2);
            b=lsort6(3);
            lsort6(2) = b;
            lsort6(3) = a;
        elseif (lsort6(3) < lsort6(4))
            a=lsort6(3);
            b=lsort6(4);
            lsort6(3) = b;
            lsort6(4) = a;
        elseif (lsort6(4) < lsort6(5))
            a=lsort6(4);
            b=lsort6(5);
            lsort6(4) = b;
            lsort6(5) = a;
        elseif (lsort6(5) < lsort6(6))
            a=lsort6(5);
            b=lsort6(6);
            lsort6(5) = b;
            lsort6(6) = a;
        else
            medianlenna(i,j) = (1/2)*(lsort6(3)+lsort6(4));
            sorted = 1;
        end
    end

elseif j == 256 %bottom row
    lsort6 = ([Y(i-1, j), Y(i, j), Y(i+1, j), Y(i-1, j-1), Y(i+1, j-1), Y(i+1, j-1)]);
    %take (2,256)          1,256          2,256          3,256          1,255          2,255
3,255
    while(sorted == 0)
        if (lsort6(1) < lsort6(2))
            a=lsort6(1);
            b=lsort6(2);
            lsort6(1) = b;
            lsort6(2) = a;
        elseif (lsort6(2) < lsort6(3))
            a=lsort6(2);
            b=lsort6(3);
            lsort6(2) = b;
            lsort6(3) = a;
        elseif (lsort6(3) < lsort6(4))
            a=lsort6(3);
            b=lsort6(4);
            lsort6(3) = b;
            lsort6(4) = a;
        elseif (lsort6(4) < lsort6(5))
            a=lsort6(4);
            b=lsort6(5);
            lsort6(4) = b;
            lsort6(5) = a;
        elseif (lsort6(5) < lsort6(6))
            a=lsort6(5);
            b=lsort6(6);
            lsort6(5) = b;

```

```

        lsort6(6) = a;
    else
        medianlenna(i,j) = (1/2)*(lsort6(3)+lsort6(4));
        sorted = 1;
    end
end

elseif i == 1%left row
    lsort6 = ([Y(i,j-1), Y(i+1, j-1), Y(i, j), Y(i+1, j), Y(i, j+1), Y(i+1, j+1)]);
    %take (1,2)          1,1      2,1      1,2      2,2      1,3
2,3
    while(sorted == 0)
        if (lsort6(1) < lsort6(2))
            a=lsort6(1);
            b=lsort6(2);
            lsort6(1) = b;
            lsort6(2) = a;
        elseif (lsort6(2) < lsort6(3))
            a=lsort6(2);
            b=lsort6(3);
            lsort6(2) = b;
            lsort6(3) = a;
        elseif (lsort6(3) < lsort6(4))
            a=lsort6(3);
            b=lsort6(4);
            lsort6(3) = b;
            lsort6(4) = a;
        elseif (lsort6(4) < lsort6(5))
            a=lsort6(4);
            b=lsort6(5);
            lsort6(4) = b;
            lsort6(5) = a;
        elseif (lsort6(5) < lsort6(6))
            a=lsort6(5);
            b=lsort6(6);
            lsort6(5) = b;
            lsort6(6) = a;
        else
            medianlenna(i,j) = (1/2)*(lsort6(3)+lsort6(4));
            sorted = 1;
        end
    end

elseif i == 256%right row
    lsort6 = ([Y(i, j-1), Y(i, j), Y(i, j+1), Y(i-1, j-1), Y(i-1, j), Y(i-1, j+1)]);
    %take (256,2)          256, 1      256, 2      256,3      255,1      255, 2
255, 3
    while(sorted == 0)
        if (lsort6(1) < lsort6(2))
            a=lsort6(1);
            b=lsort6(2);
            lsort6(1) = b;
            lsort6(2) = a;
        elseif (lsort6(2) < lsort6(3))
            a=lsort6(2);
            b=lsort6(3);
            lsort6(2) = b;
            lsort6(3) = a;
        elseif (lsort6(3) < lsort6(4))
            a=lsort6(3);
            b=lsort6(4);
            lsort6(3) = b;
            lsort6(4) = a;
        elseif (lsort6(4) < lsort6(5))
            a=lsort6(4);
            b=lsort6(5);
            lsort6(4) = b;
            lsort6(5) = a;
        elseif (lsort6(5) < lsort6(6))
            a=lsort6(5);
            b=lsort6(6);

```



```

        lsort6(5) = b;
        lsort6(6) = a;
    else
        medianlenna(i,j) = (1/2)*(lsort6(3)+lsort6(4));
        sorted = 1;
    end
end
else
    lsort = ([Y(i-1,j-1), Y(i,j-1), Y(i+1,j-1), Y(i-1,j), Y(i,j), Y(i+1,j), Y(i-1,j+1),
Y(i,j+1), Y(i+1,j+1)]);
    %take 2,2    1,1    2,1    3,1    1,2    2,2    3,2    1,3
    2,3    3,3
    while(sorted == 0)
        if (lsort(1) < lsort(2))
            a=lsort(1);
            b=lsort(2);
            lsort(1) = b;
            lsort(2) = a;
        elseif (lsort(2) < lsort(3))
            a=lsort(2);
            b=lsort(3);
            lsort(2) = b;
            lsort(3) = a;
        elseif (lsort(3) < lsort(4))
            a=lsort(3);
            b=lsort(4);
            lsort(3) = b;
            lsort(4) = a;
        elseif (lsort(4) < lsort(5))
            a=lsort(4);
            b=lsort(5);
            lsort(4) = b;
            lsort(5) = a;
        elseif (lsort(5) < lsort(6))
            a=lsort(5);
            b=lsort(6);
            lsort(5) = b;
            lsort(6) = a;
        elseif (lsort(6) < lsort(7))
            a=lsort(6);
            b=lsort(7);
            lsort(6) = b;
            lsort(7) = a;
        elseif (lsort(7) < lsort(8))
            a=lsort(7);
            b=lsort(8);
            lsort(7) = b;
            lsort(8) = a;
        elseif (lsort(8) < lsort(9))
            a=lsort(8);
            b=lsort(9);
            lsort(8) = b;
            lsort(9) = a;
        else
            medianlenna(i-1,j-1) = lsort(5);
            sorted = 1;
        end
    end
end
end
end
subplot(3, 3, 6);
image(medianlenna)
title('Median filtered Image');

```